**TWICE:**

|  |  |
| --- | --- |
| **Nombre** | RF1: Iniciar Sesión |
| **Resumen** | El sistema permite iniciar sesión |
| **Entrada** | El **correo** y la **contraseña** |
| **Salida** | Se ha iniciado sesión satisfactoriamente |

|  |  |
| --- | --- |
| **Nombre** | RF2: Mantener Sesión Abierta |
| **Resumen** | El sistema permite mantener la sesión abierta, si esta opción está activada se iniciará sesión automáticamente cada vez que se inicie el programa. |
| **Entrada** | Ninguna |
| **Salida** | La sesión se mantiene abierta. |

|  |  |
| --- | --- |
| **Nombre** | RF3: Registrar Cliente |
| **Resumen** | El sistema permite registrar un cliente en la aplicación, el email debe contener el carácter “@” y la contraseña debe tener entre 8 y 20 caracteres y contener al menos 1 carácter de las siguientes categorías: mayúsculas (A - Z) y números (0 - 9). No se pueden registrar dos usuarios con el mismo email. |
| **Entrada** | (1) El **nombre** del cliente, (2) el **email** del clientey(3) la **contraseña** del cliente |
| **Salida** | El cliente se ha registrado satisfactoriamente. |

|  |  |
| --- | --- |
| **Nombre** | RF4: Registrar Administrador |
| **Resumen** | El sistema permite registrar un administrador en la aplicación, el email debe contener el carácter “@” y la contraseña debe tener entre 8 y 20 caracteres y contener al menos 1 carácter de las siguientes categorías: mayúsculas (A - Z) y números (0 - 9). No se pueden registrar dos usuarios con el mismo email. |
| **Entrada** | (1) El **nombre** del administrador, (2) el **email** del administrador y(3) la **contraseña** del administrador |
| **Salida** | El administrador se ha registrado satisfactoriamente. |

|  |  |
| --- | --- |
| **Nombre** | RF5: Registrar Carro |
| **Resumen** | El sistema permite registrar un carro, la placa de contener tener entre 6 y 7 caracteres y tener al menos 3 letras mayúsculas y 3 números. Un usuario no puede tener dos carros con el mismo nombre |
| **Entrada** | (1) El **nombre** del carro, (2) la **placa** del carro, (3) el **tipo de combustible**, (4) el **tipo de carro**, (5) el **nivel de polarizado**. |
| **Salida** | El carro se ha registrado satisfactoriamente |

|  |  |
| --- | --- |
| **Nombre** | RF6: Registrar Moto |
| **Resumen** | El sistema permite registrar una moto, la placa de contener tener entre 6 y 7 caracteres y tener al menos 3 letras mayúsculas y 3 números. Un usuario no puede tener dos motos con el mismo nombre |
| **Entrada** | (1) El **nombre** de la moto, (2) la **placa** de la moto, (3) el **tipo de combustible** de la moto, (4) el **tipo de moto**, (5) el **cilindraje** de la moto. |
| **Salida** | La moto se ha registrado satisfactoriamente |

|  |  |
| --- | --- |
| **Nombre** | RF7: Registrar Bicicleta |
| **Resumen** | El sistema permite registrar una bicicleta, un usuario no puede tener dos bicicletas con el mismo nombre. |
| **Entrada** | El **nombre** y **color** de la bicicleta. |
| **Salida** | La moto se ha registrado satisfactoriamente |

|  |  |
| --- | --- |
| **Nombre** | RF8: Organizar Vehículos |
| **Resumen** | El sistema permite ordenar los vehículos por nombre, placa y usos (cuantas veces se ha estacionado en algún parqueadero) |
| **Entrada** | Ninguna. |
| **Salida** | Los vehículos se han ordenado por el criterio especificado. |

|  |  |
| --- | --- |
| **Nombre** | RF9: Buscar Vehículos |
| **Resumen** | El sistema permite buscar todos los vehículos que coinciden exacta o parcialmente con el nombre buscado. |
| **Entrada** | El **nombre** buscado |
| **Salida** | Todos los vehículos que coinciden exacta o parcialmente con el nombre buscado. |

|  |  |
| --- | --- |
| **Nombre** | RF10: Persistir Usuarios |
| **Resumen** | El sistema permite guardar todos los usuarios y su información, cada vez que se abra el programa, se cargarán todos los usuarios. |
| **Entrada** | Ninguna. |
| **Salida** | Los usuarios se guardan y cargan satisfactoriamente. |

|  |  |
| --- | --- |
| **Nombre** | RF11: Eliminar Vehículo |
| **Resumen** | El sistema permite eliminar un vehículo, no se puede eliminar un vehículo si no se han pagado todas sus facturas. |
| **Entrada** | El **nombre** del vehículo. |
| **Salida** | El vehículo se eliminó satisfactoriamente |

|  |  |
| --- | --- |
| **Nombre** | RF12: Pagar Recibo |
| **Resumen** | El sistema permite pagar un recibo, el precio del recibo se define con el tiempo de estancia en el parqueadero y el precio por hora para el tipo de vehículo |
| **Entrada** | Ninguna |
| **Salida** | La factura se ha pagado satisfactoriamente |

|  |  |
| --- | --- |
| **Nombre** | RF13: Cerrar Sesión |
| **Resumen** | El sistema permite cerrar sesión |
| **Entrada** | Ninguna |
| **Salida** | Se cerró la sesión satisfactoriamente |

|  |  |
| --- | --- |
| Nombre | RF14: Registrar parqueadero |
| Resumen | El sistema permite registrar un parqueadero, no pueden existir dos parqueaderos con el mismo nombre para el mismo usuario. |
| Entrada | (1) El **nombre**, (2) la **dirección** del parqueado, (3) la **ciudad** del parqueadero, (4) el **departamento** del parqueadero, (5) el **país** del parqueadero, (6) el **email** del parqueadero, (7) la **información** del parqueadero (comentarios adicionales sobre el parqueadero) y (8) los **precios** por hora del parqueadero |
| Salida | El parqueadero se ha registrado satisfactoriamente |

|  |  |
| --- | --- |
| Nombre | RF15: Importar parqueadero |
| Resumen | El sistema permite importar un parqueadero de un archivo, no pueden existir dos parqueaderos con el mismo nombre para el mismo usuario. |
| Entrada | La **ruta** del archivo con la información del parqueadero |
| Salida | El parqueadero se ha registrado satisfactoriamente |

|  |  |
| --- | --- |
| Nombre | RF16: Ordenar parqueaderos |
| Resumen | El sistema permite ordenar los parqueaderos por nombre, dirección y precio |
| Entrada | Ninguna |
| Salida | Lo parqueaderos se han ordenado satisfactoriamente |

|  |  |
| --- | --- |
| Nombre | RF17: Buscar parqueaderos por nombre |
| Resumen | El sistema permite buscar todos los parqueaderos que coinciden exacta o parcialmente con el nombre buscado. |
| Entrada | El **nombre** buscado |
| Salida | Los parqueaderos que coinciden exacta o parcialmente con el nombre buscado |

|  |  |
| --- | --- |
| Nombre | RF18: Buscar Reportes |
| Resumen | El sistema permite buscar todos los reportes de un usuario. |
| Entrada | Ninguna |
| Salida | Todos los reportes del usuario |

|  |  |
| --- | --- |
| Nombre | RF19: Registrar espacios |
| Resumen | El sistema permite registrar espacios en un parqueadero. |
| Entrada | La **cantidad** de espacios y el **tipo** de espacio |
| Salida | Se han registrado satisfactoriamente los espacios |

|  |  |
| --- | --- |
| Nombre | RF20: Generar Url del Mapa |
| Resumen | El sistema permite generar una url en google maps con la dirección del parqueadero |
| Entrada | Ninguna |
| Salida | Se ha generado la url satisfactoriamente |

|  |  |
| --- | --- |
| Nombre | RF21: Contar puestos vacíos de un parqueadero |
| Resumen | El sistema permite contar cuántos espacios vacíos hay para un tipo específico de vehículo en un parqueadero |
| Entrada | El **tipo** de vehículo. |
| Salida | El número de espacio vacíos de ese tipo de vehículo en el parqueadero |

|  |  |
| --- | --- |
| Nombre | RF22: Eliminar Parqueadero |
| Resumen | El sistema permite eliminar un parqueadero por el nombre |
| Entrada | El **nombre** del parqueadero |
| Salida | Se ha eliminado el parqueadero satisfactoriamente |

|  |  |
| --- | --- |
| Nombre | RF23: Ingresar Vehículo Parqueadero |
| Resumen | El sistema permite ingresar un vehículo en un parqueadero, no se puede parquear un vehículo que tenga facturas pendientes. Una vez se haya ingresado el vehículo, se generará una factura y un reporte con la fecha y hora de ingreso del vehículo. |
| Entrada | Ninguna |
| Salida | El vehículo se ingresó al parqueadero satisfactoriamente |

|  |  |
| --- | --- |
| Nombre | RF24: Sacar Vehículo Parqueadero |
| Resumen | El sistema permite sacar un vehículo de un parqueadero, cuando se saca el vehículo se genera una factura con la fecha y hora de salida del vehículo y el precio que debe pagar. |
| Entrada | Ninguna. |
| Salida | Se sacó el vehículo del parqueadero satisfactoriamente |

**Requerimientos no funcionales:**

* Toda la información debe ser persistente.
* Una parte de la persistencia debe ser manejada a través de archivos de texto.
* Una parte de la persistencia debe ser manejada a través de serialización.
* Usar 4 excepciones propias del API de Java.
* Crear 3 excepciones propias.
* Implementación de pruebas unitarias automáticas para todos los métodos del modelo.
* Incluir dos implementaciones propia de algoritmos de ordenamiento con criterios diferentes (burbuja, selección, inserción).
* Incluir dos implementaciones propias de un algoritmo de búsqueda binaria usando un criterio diferente para cada implementación.
* Incluir dos implementaciones de listas enlazadas.
* Incluir 4 herencias (que no sean del mismo padre).
* Incluir 3 interfaces.
* Incluir 2 implementaciones propias de árboles binarios.
* Incluir 4 métodos recursivos en estructuras naturalmente recursivas.
* Dibujar un gráfico con dos primitivas.
* Usar 3 hilos diferentes al principal.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores Entrada** | **Resultado** |
| MotorVehicle | validatePlate | Un vehículo vacío. | Comprueba las placas “A1B2C3”, “A1B2C3D”, “!@#$%^”, “ASDF1234”. | Recibe solo las 2 primeras. |
| MotorVehicle | comparePlate | Un vehículo vacío de placa “JHO907”. | Se compara con los vehiculos “DAU345”, “MIA820”, “JHO907”, bicicleta. | Es mayor, es menor, es igual y es menor. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores Entrada** | **Resultado** |
| Vehicle | addBill | Un bicicleta vacía. | Se añade un recibo de los parqueaderos “Parqueadero Nayeon”, “Parqueadero Sana” | Están almacenados los dos recibos. |
| Vehicle | unpaidBills | Un bicicleta con tres recibos. El de la mitad está pago. | Se sacan los recibos no pagos. | Los recibos de los extremos. |
| Vehicle | billsSize | Un bicicleta vacía y una con tres recibos. |  | 0 recibos y 3 recibos. |
| Vehicle | compareTo | Un bicicleta de nombre “chaeyoung”. | Se compara con “ariza”, “dahyun” y “chaeyoung”. | Es mayor, es menor y es igual. |
| Vehicle | compare | Una bicicleta normal. | Se compara bicicleta de recibos 0 y 1, 2 y 1, 1 y 1. | Es menor, es mayor y es igual. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores Entrada** | **Resultado** |
| Report | addReport | Un reporte vacío de email “twice..”. | Se añaden los reportes “yooje..”, “nosanano..”, “twice..”, “yooje..”, “nosanano..”. | Derecha dos “yooje”, izquierda dos “nosanano..” y centro otro “twice..”. |
| Report | addSameEmailReport | Un reporte vacío de email “twice..”. | Se añaden 2 reportes “twice...” | Se añaden los reportes. |
| Report | searchReports | Un reporte con reportes a los lados y centro “twice..”,”nosanano..”,”yoo..”. | Se buscan los reportes de esos nombres y tambien “momo..”. | Se encuentran todas excepto “momo..”. |
| Report | compareTo | Un reporte vacío de email “twice..”. | Se compara con “yoo..”, “nosanano..”, “twice..”. | Es menor, es mayor y igual. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores Entrada** | **Resultado** |
| Slot | insertVehicle | Un espacio de tipo vehículo motorizado. | se añade un carro, una moto. Vacío, se añade una bicicleta. | Se añade el carro con su recibo y todo. No deja añadir la moto y bici. |
| Slot | removeVehicle | Un espacio precio 200 con vehículos de una hora pasada. | Se saca el vehículo. | El precio es aproximado de 200 y se añadió datos recibo. |
| Slot | calculatePrice | Un espacio precio 1000, 500, 200 con vehículos de una media hora, 2 horas y 1 hora pasada. |  | Precios son de 500, 1000, 200. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores Entrada** | **Resultado** |
| Parking | addSlot | Un parqueadero vacío. | Se añade espacios bici 1, moto 2, carro 1. | Se añadieron todos. |
| Parking | addReport | Un parqueadero vacío. | Se añade reporte “nosana..”, “eagle..”, “tzuyo..”, “nosanano..”. | Se añadieron todos. |
| Parking | insertVehicle | Un parqueadero con 4 slots. | Se añaden 2 vehículos a slots diferentes. | Se anaden con todo. |
| Parking | removeVehicle | Un parqueadero con 4 slots, 2 ocupados. | Se remueve el carro. | Espacios están vacíos y se completaron los reportes. |
| Parking | searchReport | Un parqueadero sin reportes y otro con reporte “nosana..”. | Se busca reporte “eagle..”, “nosana..”. | No se encuentra y se encuentra. |
| Parking | searchLastReport | Un parqueadero sin reportes y otro con reporte “nosana..”. | Se busca reporte “eagle..”, “nosana..”. | No se encuentra y se encuentra. |
| Parking | searchSlotOwnerVehicle | Un parqueadero vacío y uno con un vehículo parqueado. | Se buscan. | No se encuentra y se encuentra. |
| Parking | isEmpty | Un parqueadero vacío, un parqueadero con espacios vacíos y uno con espacios llenos. |  | Vacio, vacio y lleno. |
| Parking | calculateAverage | Un parqueadero de precios {1000, 500, 200} y otro {200, 500, 200} |  | Promedio 566.6 y 900. |
| Parking | emptySlotsQuantity | Un parqueadero con 4 espacios. carro, moto, bici y vehículo motorizado. |  | 2 carro, 2 moto, 1 bici y 1 vehículo motorizado. |
| Parking | GenerateMapUrl | Un parqueadero de dirección “Col, Val, Cali, cl. 15 #121-25”. | Se genera. | URL igual a la de google maps. |
| Parking | compareTo | Parqueadero de nombre “Twice”. | Se compara con “Dahyun”, “Momo” y “Twice” | Es mayor, es menor y es igual. |
| Parking | compare | Parqueadero de dirección “Cali” | Se compara con “Bogotá”, “Medellín” y “Cali” | Es mayor, es menor y es igual. |
| Parking | compareByAverage | Parqueadero precio promedio 566.6 | Se compara con 400, 600 y 566.6 | Es mayor, es menor y es igual. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores Entrada** | **Resultado** |
| Client | addVehicle | Un cliente vacio. | Se añaden vehiculos a traves de archivos de texto. Algunos mal y repetidos. | Se añaden solo los primeros. |
| Client | addCar | Un cliente vacio. | Se añaden carros. Algunos con placas malas o repetidos. | Se añade solo 1. |
| Client | addMotorcycle | Un cliente vacio. | Se añaden motos. Algunos con placas malas o repetidos. | Se añade solo 1. |
| Client | addBicycle | Un cliente vacio. | Se añaden bicicletas iguales. | Se añade solo 1. |
| Client | deleteVehicle | Un cliente con 4 vehículos. | Se borra uno y otros que no existen . | Tiene 3 vehículos. |
| Client | searchVehicle | Un cliente 4 vehículos. | Se busca uno y otros que no existen . | Lo encuentra y no lo encuentra. |
| Client | searchVehicles | Un cliente con 4 vehículos, 2 comienzan por “ji”. | Se busca “ji”. | Se encuentran los 2. |
| Client | searchMotorVehicles | Un cliente 4 vehículos 3 con placa. | Se busca uno y otros que no existen . | Lo encuentra y no lo encuentra. |
| Client | sortVehiclesByName | Un cliente 4 vehículos. | Se organizan por nombre. | Están ordenados. |
| Client | sortVehiclesByUses | Un cliente 4 vehículos. | Se organizan por uso. | Están ordenados. |
| Client | sortVehiclesByPlate | Un cliente 4 vehículos 3 con placa. | Se organizan por placa. | Están ordenados. |
| Client | load | Un cliente vacio. | Se cargan 3 archivos y uno corrupto. | Crea los 3 vehículos. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores Entrada** | **Resultado** |
| Owner | addParking | Un dueño vacío | Se le agregan 5 parqueaderos y uno de esos ya existía | Se agregaron 4 parqueaderos |
| Owner | addParkingPath | Un dueño vacío | Se carga desde un archivo dos veces el mismo archivo | Se agrega un parqueadero |
| Owner | deleteParking | Un dueño con 3 parqueaderos | Se eliminan dos parqueaderos | El dueño tiene dos parqueaderos |
| Owner | searchParkings | Un dueño con 3 parqueaderos | Se busca con la palabra “Mo” | Encuentra los parqueaderos con los nombres “mona” y “Momo” |
| Owner | searchRight | Un dueño con 3 parqueaderos | Se busca con la palabra “Mo” a la derecha | Encuentra el parqueadero con el nombre “Momo” |
| Owner | serchLeft | Un dueño con 3 parqueaderos | Se busca con la palabra “Mo” a la izquierda | Encuentra el parqueadero con el nombre “mona” |
| Owner | checkAlreadyExist | Un dueño con 3 parqueaderos | Añaden 2 parqueadero (Uno de estos ya estaba pago) | Añade un parqueadero |
| Owner | sortParkingsByName | Un dueño con 3 parqueaderos | Se ordenan los parqueaderos por nombre | Los parqueaderos están ordenados por el nombre |
| Owner | sortParkingsByAddress | Un dueño con 3 parqueaderos | Se ordenan los parqueaderos por nombre | Los parqueaderos están ordenados por el nombre |
| Owner | sortParkingsByPrice | Un dueño con 3 parqueaderos | Se ordenan los parqueaderos por precio | Los parqueaderos están ordenados por el precio |
| Owner | load | Un dueño vacío | Se cargan desde un archivo plano dos parqueaderos ( uno de ellos desde un archivo inválido) | Se agregó un parqueaderos |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores Entrada** | **Resultado** |
| User | addUser | Un usuario vacío | Se agregan 5 usuarios (3 que ya existían) | Se agregaron 3 usuarios |
| User | searchUser | 3 usuarios | Se buscan 4 usuarios (uno que no existe) | Se encontraron 3 usuarios |
| User | verifyPassword | Un usuario vacío | Se prueban 5 contraseñas, “Password6”, “Pass6”, “PasswordPasswordPassword6”, “password6”, “PasswordSix” | Solo 1 es válida, “Password6” |
| User | verifyEmail | Un usuario vació | Se prueban 2 correos “[nosananolife@gmail.com](mailto:nosananolife@gmail.com)” y “nosananolifegmail.com” | Solo “[nosananolife@gmail.com](mailto:nosananolife@gmail.com)” es valido |
| User | encrypt | Un usuario vacío | Se encripta “Chae ” | El texto encriptado es “009101097104067” |
| User | decrypt | Un usuario vacío | Se desencripta “009101097104067” | El texto desencriptado es “Chae ” |
| User | compareTo | Un usuario vacío | Se compara “Momo” con “Nayeon”, “Chaeyoung” y ”Momo” | Es mayor, es menor y es igual. |
| User | getAllParkings | 3 usuarios | Se obtienen todos sus parqueaderos | Son todos sus parqueaderos |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores Entrada** | **Resultado** |
| App | addClient | Una app vacía | Se agregan 2 clientes | Los dos clientes se agregaron |
| App | addOwner | Una app vacía | Se agregan 2 dueños | Los dos dueños se agregaron |
| App | logIn | Una app con 3 usuarios | Se intenta iniciar sesión con 3 correos y contraseñas distintos | Se inicia sesión con uno de ellos |
| App | searchUser | Una app con 3 usuarios | Se buscan 3 usuarios (uno que no existe) | Se encuentran 2 usuarios |
| App | sortParkingsByName | Una app con 3 usuarios y 6 parqueaderos | Se ordenan por nombre "Jeongyeon", "Momo", "Sana", "Mina", "Chaeyoung", "Jihyo", "Tzuyu" y "Dahyun" | Los parqueaderos quedan "Chaeyoung", “Dahyun”, “Jeongyeon”, “Mina”, “Momo” y “Tzuyu” |
| App | sortParkingsByAddress | Una app con 3 usuarios y 6 parqueaderos | Se ordenan por dirección | Están ordenados por dirección |
| App | sortParkingsByPrice | Una app con 3 usuarios y 6 parqueaderos | Se ordenan por price | Están ordenados por precio |
| App | load | Una app con 3 usuarios | Se carga el usuario actual desde un archivo de text | El usuario del archivo de texto es igual al usuario actual |
| App | getAllParkings | Una app vacía | No tiene parqueaderos, después se carga un app con 3 usuarios y 6 parqueaderos. | Hay 0 parqueaderos, después hay 6 parqueaderos |